Blockchain-Based Platform Architecture for Industrial IoT

Nikolay Teslya, Igor Ryabchikov
SPIIRAS
St.Petersburg, Russia
teslya@iias.spb.su, i.a.ryabchikov@gmail.com

***Abstract*— The development of robotics, the Internet of Things concept, big data processing techniques, automation, and distributed digital ledgers leads to the fourth industrial revolution. One of the main issues of new industry is interaction between the "smart factory" components both internally and with other factories based on the Internet of Things. This interaction should provide trust between the participants of the Internet of Things; control over the distribution of resources (such as maintenance time, energy, etc.) and finished products. The paper describes one of the possible ways of integrating Internet of Things and blockchain technologies to solve these issues. For this purpose, an architecture has been developed that combines Smart-M3 information sharing platform and blockchain platform. One of the main features of the proposed architecture is the use of smart contracts for processing and storing information related to the interaction between smart space components.**

I. Introduction

According to various estimates, from two to three industrial revolutions have already been committed in industry by now. The first is characterized by the transition from manual to machine work (industrial revolution, 1760 — 1840) and the second — by the development of conveyor production and electricity (power revolution, late 19th century — early 20th century). The third industrial revolution, also known as computer or digital revolution, is characterized by the application of information technology and partial robotization of production (1960 — now). There is still no final opinion about when it has ended. At the same time, there are many publications that consider to the approach of the fourth industrial revolution, also known as transfer to the "Industry 4.0" [1], [2]. Its approach is associated with the development of robotics, the introduction of the Internet of Things (IoT) concept, the development of big data processing techniques, automation, and distributed digital ledgers. The use of these technologies in production makes it possible not only to automate it, but also to create "smart factories", which can be viewed as cyberphysical systems that have complete autonomy and awareness of the production process, and are able to interact within the physical and virtual worlds with other industries. This approach allows making the production configurable, adaptable to customer needs, which in turn allows to produce highly customized products, without the need for a deep reconfiguration of the production base.

In the context of the Industry 4.0, one of the main issues is the organization of interaction between the "smart factory" components both internally and with other factories. This problem is usually solving by using the IoT concept, which allows uniting many components into a single information space and providing information exchange between them. Regarding to the industry considers the concept of Industrial Internet of Things (IIoT), which is the use of IoT for the interaction of physical, virtual and social industrial components in a single information space also knows as smart space. At the same time, production becomes decentralized and several problems appear, among which the following can be highlighted: the need to provide interoperability between components in the smart space and between smart spaces; trust between the participants of the information space; control over the distribution of resources (such as maintenance time, energy, etc.) and finished products.

To provide the interoperability between the smart space components, the ontology and ontology matching mechanism can be used. Such approach is already described and used in the number of projects, e.g. [3], [4].

The solution of the problems of trust between components can be solved with the help of digital signature [5] and access control [6] mechanisms. Control over resources distribution and finished products can be carried out using a database accessible to all components. These solutions are quite complex and require the deployment of complex infrastructure to provide fault tolerance, performance and availability. At the same time, there is an active development of blockchain technology, which provides a simpler solution for the problems presented above.

This paper describes integration of IoT and blockchain [7] technology to solve the tasks of providing trust between the components of the production network and controlling resources distribution, as well as finished products distribution. For this purpose, an architecture has been developed that combines one of the platforms for the organization of the IoT — Smart-M3 and the blockchain. One of the main features of the proposed architecture is the use of smart contracts for processing and storing information related to the interaction between smart space components.

The rest of the paper is structured as follows. Section 2 describes common scenarios and specifics of IIoT. Section 3 provides problem statement with analysis of lacks existing platforms and specification of requirements to IIoT platform. Section 4 provides brief description of blockchain and smart contracts. Section 5 describes an architecture of IIoT platform built based on integration of Smart-M3 and Hyperledger blockchain platform. Section 6 provides discussion over advantages and disadvantages of proposed platform.

II. Application scenarios of the industrial internet of things concept

In [8] several scenarios of Industrial Internet of Things are described:

1) Production on demand. It is possible to create marketplaces of production services that will accept and automatically serve orders from buyers to produce highly customizable goods (for example 3D printing or computer numerical control). Production components would be smart devises tracking orders transmitted through a blockchain network.
2) Tracing of goods through supply chains. The creation of digital twins of goods which would store information about goods life cycle — from what details they were produced, who and when owned them, information about repairs etc. Thanks to this information it is possible to track the position of the product in the supply chain, identify the products that were produced from the batch of defective parts, confirm the product license (that it is not fake or stolen) etc. The creation of digital twins with similar goals is also described in papers [9], [10], [11]. The example of a supply chain schema is described in paper [9] and presented in Fig. 1.
3) The organization of automatic interaction between production machines — the exchange of messages about the readiness of the product to the next stage of production. Herewith production can be decentralized — different stages of processing can be carried out by different responsible enterprises.

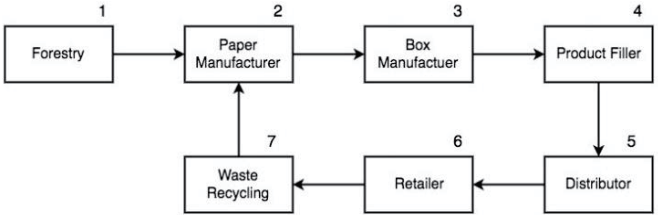


Fig. 1. Part of the manufacturing supply chain for a cardboard box [9]

In paper [12] the application of the IoT concept and smart space for the organization of work for lenses assembly line is described.

In [2] the advantages of the fourth industrial revolution are described (among which there are high customization of goods for the users, more efficient use of resources, reduction in the production time and so on) as well as technical capabilities that make it possible (automation and mechanization, the digitalization of the components of production (machines, resources, etc.)).

III. Problem statement

A. Overview of Existing IoT platforms

Now, there are known several architectures of platforms for organizing the common smart space between participants, for example, Smart-M3 [13], Smart Platform [14], Smart-X [15], the architecture of a distributed platform to manage products information [16], but they all have shortcomings that prevent their usage in the IIoT. This work is based on the Smart-M3 platform that is open sourced and supported by FRUCT community. The architecture of the platform is shown in Fig. 2.

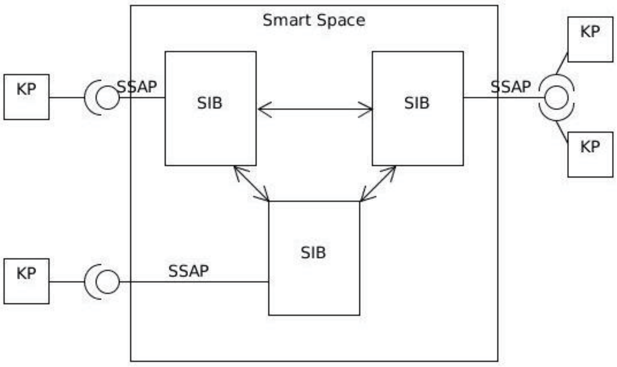


Fig. 2. Smart-M3 system architecture

The main component of the platform is Semantic information broker (SIB), which stores information and services requests of participants (knowledge processors (KP)) of smart space by smart space access protocol (SSAP). As a format for the presentation of information, RDF is used. The SSAP allows recording, deleting RDF statements, as well as querying information and subscribe to the appearance of the required information. Patterns of the required information can be described in the form of RDF triplets or by more expressive query languages (for example, SPARQL).

The smart space may consist of one or, as intended, a set of connected SIBs. The existing implementations support the formation of smart space by only one SIB, but the support of multiple interacting SIBs can be implemented in the future. In [17] a variant of such interaction is described. All information is distributed between SIBs, thereby scaling provided. Participants can send requests to any SIB, while all information of the smart space distributed among all SIBs will be available to them. When a SIB receives a query from KP, it extends it to all other SIBs. Each executes the received query and returns a reply to the SIB-initiator, which aggregates results and transfers to the KP. But there are restrictions – the information visible to KPs, describes as following:

$$\bigcup_{\beta} \Delta(i(\beta)), \tag{1}$$

where β — set of SIBs, Δ — deductive closure over the space, and $i(\beta)$ — being the information contained in each SIB [17].

Thus, KP only has access to the information that can be inferred locally to a SIB. Because of this, some complex queries (for example, SPARQL) can produce an incomplete result, since they will be executed independently locally to each SIB.

The simplest solution for organizing a common smart space for distributed participants is to deploy Smart-M3 platform as a centralized cloud service. The provider of such service may be one of the participants or a third party.

However, this approach has several significant shortcomings, one of which is the low security — in this architecture, there is a need for trust of participants of the smart space to the service provider. Participants can publish their information and query information of others, but there is no guarantee that certain information was published by a certain participant, and that all information published by that participant is still available. The platform provider can return any information in response.

Another shortcoming is the low fault tolerance. Disruption of the information system of one participant should lead only to the disruption of business processes in which it takes active part, but the other participants should be able to continue they collaboration. In this architecture, all participants depend on the central provider. However, in the IIoT participants of the smart space can be large enterprises interacting with each other and, thereby, organizing distributed business processes. Trust to third parties is unacceptable due to the huge risks inherent in large companies.

Another solution based on Smart-M3 platform is a creation of distributed smart space. Each participant could deploy his own SIB, to which only his own KPs would connect, and which would store only their information. Since the certain participant is an information provider himself, without any third-party provider, then this architecture could satisfy the **security** requirement. The absence of a single point of failure in this case would allow, in the case of the failure of SIB of one enterprise, to continue the work of others — the requirement of **fault tolerance**. If the decision proposed in [17] were taken as the SIB interaction mechanism, then some modifications would be required to satisfy the indicated requirements. For example, the work [17] describes the possibility of a SIB transit connection. In the case of this paper, it is necessary to connect SIBs directly to each other without intermediaries.

However, the considered distributed architecture also has drawbacks. The result returned by a certain SIB says that this set of statements is the knowledge of the corresponding participant at a given time. There is no guarantee that the statements were effective earlier (they were published earlier and did not change from this moment). Moreover, in the case of the SIB party outage, all its information could be no longer available. This can disrupt the work of other participants. Let us consider one of the most common scenarios of the Industrial Internet of Things - the supply chain. Keeping comprehensive information about products in the common public smart space can allow them to trace their life cycle: the ways of production, storage and transportation, therefore, draw conclusions about their quality. All participants during the product life cycle (from enterprises that producing raw materials to the end users) can be consumers of information about products, and information should always be available to them. At the same time, such information must be unchangeable even for its author. For example, if a violation is detected after production, for example, a batch of products was produced from poor-quality material, and it may be advantageous for the enterprise to hide information about which goods belong to this party. Thus, the requirement of **durability** can be singled out.

This architecture also does not guarantee that for a single request from different participants, a certain SIB will return the same information. Some information should be the same for all participants, for example, to determine the status of a distributed business process by all its participants. A participant can publish information about the transfer of an ownership of certain virtual object to another participant in the fulfilment of the transaction. For the buyer, the seller can display that the object was transferred to him, but for others — that the seller still owns this object, thereby the buyer cannot confirm the rest his ownership. This requirement can be called as a **public access possibility** to the information.

Although, this architecture can be suitable for organizing communication between participants in real time or for publishing information that is useful only in case of it is trustworthy to its author (for example, a certain company can distribute e-tickets that will be claimed only at its own purpose).

Participants' interactions can occur in accordance with distributed business processes. To use the smart space for coordination of joint actions, it is necessary to determine the general state of the business process and the necessary transition that is the **opportunity to come to a consensus**. This is achievable, for example, if it is possible to determine the sequence of statements of participants related to the business process. An example of such a business process can be the accounting and transfer of ownership of the object. Transfer of ownership can be conducted by publishing the appropriate approval of the owner of the object. The current owner can be determined by examining the ownership transfer chain.

The drawback of the distributed architecture presented in [17] is also the limitations of logical input and complex search by the local base of a single SIB, which does not allow creating search constraints that include knowledge of different participants. For example, it is impossible to identify all the final products that were produced from a certain lot of defective parts by a single request, if production was carried across several factories. For this search, it is needed to provide series of separate queries.

Other platforms have the same drawbacks. In [14], a platform is proposed for organizing a smart general-purpose space, but using a central component for messages distribution. In [15], the architecture of the platform is proposed, primarily designed for monitoring heterogeneous systems. It also uses centralized nodes, built in a hierarchy. In [16] the architecture of the platform for publishing information about goods and its effective search, which can be used in supply chains, is proposed. The architecture is similar to the distributed smart space based on Smart-M3, described above, and has the same drawbacks. The difference is the presence of the second type of node in the peer-to-peer network, which, in effect, performs indexing (and ordering) of statements about the goods of different participants. Providers of these nodes are manufacturers of the corresponding goods, and it is assumed that they are interested in the availability of information about their products. However, this decision violates the requirements of situation described in the paper.

B. Platform requirements for the IIoT

Summarizing the previous paragraph, it is possible to designate the requirements that should be satisfied during the development of the platform for the organization of smart space for Industrial Internet of Things:

1) **Security** – guarantee that a certain participant published the information, and this is all information that he published. That is, third parties should not be able to forge or hide part of the information.
2) **Fault tolerance** – the disruption of the information system of one participant should lead only to the disruption of business processes in which he participates. The work process of the others should not be affected.
3) **Durability** – being published once, the information should remain accessible to all participants.
4) **Public access possibility** – some information should be viewed the same for all participants.
5) **The possibility of consensus** – interactions of participants can take place in accordance with distributed business processes. For using smart space to coordinate joint actions, it is necessary to determine the single general state of the business process and the necessary transition.

Considering scenarios of the IIoT it is possible to single out the following additional requirements to the platform:

1) The ability to filter the requested information by its author. In the IIoT participants of the common smart space can be independent enterprises that must coexist in it and have equal rights for publishing information. For example, everyone should have the possibility to say: “It is raining in Moscow”, but in a certain situation for certain questions not all can be trusted sources. Therefore, one of the requirements for smart space is the ability to filter the requested information by its author. For example, “What is the weather like in Moscow now, according to the state weather service?”
2) The opportunity to mark the information as irrelevant and to filter it in queries. The information published in the smart space may lose relevance (for example, an information about a current owner of the product), but the information should not be removed based on the durability requirement. Therefore, a mechanism/agreement is needed according to which the information can be marked as irrelevant, as well as a mechanism of filtering such information in queries, allowing to specify an expression like “Who is a current owner of this product?”

IV. BLOCKCHAIN AND SMART CONTRACTS

The requirements presented in the previous section can be satisfied by using the blocking technology and smart contract concept.

A. Blockchain

Blockchain was originally considered as a distributed transaction ledger for keeping records of operations with the cryptocurrency [7], but it can also be used for other purposes. It is a chain of transaction blocks containing a header and a list of transactions. The header specifies its own hash, hash of the previous block, hash of transactions and additional service information. To calculate transaction hashes, the Tiger Tree Hashing algorithm is using. Thus, due to linking with hash of the previous block and calculation of the total hash of several transactions in this technology, the data in block cannot be changed.

In the simplest case, a transaction is a record that specifies the operation id and type, the operation itself, and the users participating in the specified operation [18]. For each user, an open-private key pair is usually formed, which is used to sign a transaction to unambiguously establish the ownership of the operation. On the next step, to form a block in the blockchain, a hash function is calculated over all transaction information, and the hash value is then used to calculate the hash of the block. Thus, information about the transaction becomes also unchangeable and ensures the safety and durability of data storage in the blockchain.

It should be noted that the transaction ledger is designed to be distributed. This means that each blockchain user has access to the entire transaction log and can check any entry that it contains. Thus, it is possible to achieve a common consensus when carrying out operations, since each user can check the hash of the new block and determine the correctness of the transaction.

The built-in consensus mechanism allows to organize all published information, so that any protocols can be implemented on top of the blockchain, and any distributed business processes can be managed. For example, ownership of virtual resources with support for their transmission and atomic exchange can be realized. To do this, the following form of statements can be used:

1) "Participant X now owns resource A";
2) "I propose to exchange resource A to B from participant X (or any other, or from the list)";
3) "I agree to the exchange of resource B to A from participant Y";
4) "I cancel the offer of exchange of resource A to B from participant X".

Statement “1” can be used for the transmission and creation of a resource. For atom exchange, statement “2” can be used and followed by “3” or “4” in case of cancel the offer. It should be noted that users could publish statements without having the necessary resource. Such statements should be ignored, and it is necessary to track who owns what resources, considering the entire history of the transactions to determine the invalid statements.

To date, the blockchain is mainly used as a basis for cryptocurrencies, for example, in platforms like Blockchain, Ethereum, LiteCoin, etc. However, during recent years there have been published works, the main purpose of which is the investigation of blockchain usage in other areas, such as supply chain [9], medicine [19], and IoT [11].

B. Smart Contract

The idea of smart contract was first proposed in 1994 by Nick Szabo, even before the development of blocking technology. N. Szabo has defined a smart contract in the following way: "A smart contract is a set of promises, specified in digital form, including protocols within which the parties perform on these promises." [20]

Smart contract within blokchain technology can be viewed as a decentralized application available to all users of the blokchain. Due to the use of the Turing-complete language, the description of the contract code allows implementing of rather complex algorithms. At the same time, it is mandatory to have conditions under which the contract must be executed, and the list of actions assigned to the submitted conditions. All conditions of a smart contract must have a mathematical description and a clear execution logic. In this regard, the first smart contracts in the chain of blocks are created to formalize the simplest relationships, and consist of a small number of conditions. Users sign a contract by using their open-private key pairs and send them in a transaction that is written to the chain of blocks. After signing by the parties, the smart contract comes into force. To ensure the automated performance of contract obligations, an environment of existence is required that allows fully automating the execution of contract items. This means that smart contracts can only exist within an environment that has unrestricted access to executable code of smart contract objects. Having unimpeded access to the objects of the contract, the smart contract monitors the specified conditions of achievement or violation of the points and makes independent decisions based on the programmed conditions. Thus, the main principle of a smart contract is the complete automation and reliability of the performance of contractual relations between people.

Some blockchain platforms that provide environment for smart contracts are listed below:

1) Bitcoin [7]. It should be noted that this platform uses not a Turing-complete language and strict restrictions on the format of the contract
2) Ethereum [21]. Contracts can be created using the internal program language – Solidity.
3) Hyperledger Fabric [22]. Platform leverages container technology to host smart contracts called "chaincode" that comprise the application logic of the system.

V. Platform architecture for IIoT

A. Smart Space Over a Blockchain

A Smart Contract over the blockchain can be developed to create a platform for the IIoT that will satisfy the specified requirements. It will provide the following functions:

1) Accept the information of participants and identify the smart space to which the published information relates.
2) Check the accepted syntax of the published information.
3) Keep the state of the smart space and update it by participants request provided that the accepted rules of consistency will be observed.

The provided interface of the Smart Contract will have a single method that takes two parameters:

1) A description of the changes — adding/removing (by marking as irrelevant) information;
2) A description of the smart space state that must act for changes to be applied.

The first parameter specifies the changes themselves. The seconds parameter is necessary for making changes provided that the smart space has a certain state (contains certain information). As it was noted before, the participants can form distributed business processes, using the smart space to account and conduct these processes — to publish information that initializes certain transitions. The state of the business process can be determined by examining the sequence of statements of its participants. One of the blockchain characteristics is the possibility of rolling back the state using a certain consensus algorithm for example Prof-of-Work in Ethereum [21] and Bitcoin [7]. Thus, a situation is possible when the participant of the business process, on the base of the published information (a certain state of a BP), can send a request for publishing information that initiates a new transition, after which the state can be rolled back. When forming a new state, the participants requests can be executed in a different order or even be discarded. Such situation can disrupt the business process and undesirable transition may occur. As an example, the production on demand scenario can be considered. The firm takes orders to produce some goods. For this purpose, it may be needed to order some components or materials from other firms. So, after the firm receives the order, it must send the relevant orders to the suppliers. If an alternative blockchain block takes place, the transaction of the customer's order can be discarded, but the transaction of the ordered components or materials can be accepted. If the customer does no repeat his order, the ordered components may be unnecessary. A possible solution of the problem is a conditional update of the state that does not allow the publication of information in the wrong order or without the necessary basic information.

Also, the reason for using this function may be the need to publish certain statements only if there is a current state that may change from the time the request is sent to the time of its processing. Other applications of this function are described below.

Both parameters can be empty. In this case the method will return true if the changes have been made and false — otherwise. Changes will be made if the condition described as the second parameter are satisfied, and the changes, described as the first parameter are correct. Changes are considered as correct if they satisfy the accepted syntax and do not break the rules of consistency.

The published information will have an identifier of its author, a serial number and a relevance mark as a certain meta-information. This meta-information will be available through a standard query mechanism with the ability to filter information by it. Thus, it is possible to filter information by authors, publication time and relevance. The serial number should be composite having in the upper digits the block number in which this information was published and in the lower digits – the number of the information unit in the block. So, it is possible to define the complete order of statements, and if the blockchain uses the blocks creation policy with the certain periodicity (for example, Ethereum with period of 17 seconds) it is possible to determine approximate time of statements publication.

As it was mentioned before, for implementation of additional protocols over an ordered list of published statements, for example, ownership and transfer of resources, it is necessary to keep a certain state and track its change according to generally accepted rules in accordance with published statements. A possible way to implement this mechanism is to use Smart Contracts. This mechanism allows distribution and applying protocol rules without going beyond the blockchain. Smart Contracts created with this purpose will implement methods in which appropriate checks and state updates will be made. State changes will be reflected in the smart space by means of adding/deleting relevant information. For this Smart Contracts of protocols will invoke the described method of the Smart Contract of the smart space.

As an example, consider a possible implementation of a Smart Contract for ownership organization and transferring of resources. The contract can have the following methods:

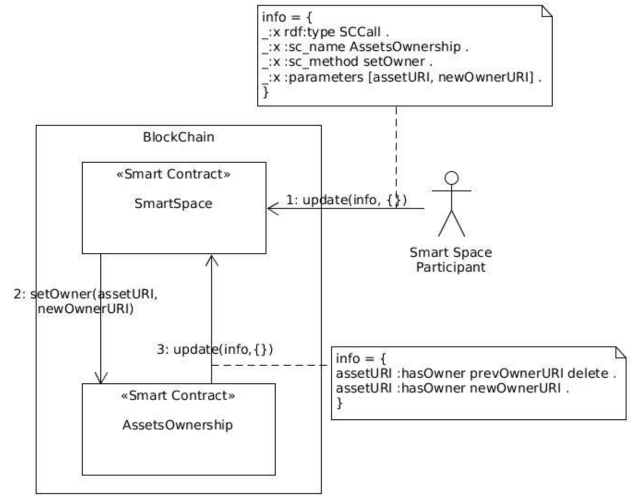
1) setOwner(resourceURI, ownerURI) — transfers resources to a new owner;
2) offerTrade(offeredResourcesURIs, requredResourcesURIs) — exposes an offer for the resources exchange;
3) commitTrade(offeredResourcesURIs, requiredResourcesURIs) — makes an exchange according to the existing offer;
4) cancelOffer(offeredResourcesURIs, requiredResourcesURIs) — cancels the offer for the resources exchange;

In case of resources transfer, the proposed Smart Contract can publish statements like “participant X owns resource A” in the smart space, which will have the identifier of the Smart Contract as a reference to the author (Smart Contracts in relation to smart space will consider equal to other participants). Thus, participants of the smart space do not need to verify the validity of these statements independently. All they need is to track the publication of the statements such as «participant X owns resource A» the author of which is the Smart Contract, and authenticity will be guaranteed.

Invocations of Smart Contracts methods are made either directly through the provided transaction initiation interface of the blockchain, or through the methods of other Smart Contracts. To simplify the use of custom protocols the Smart Contracts methods invocation through the Smart Contract of the smart space should be implemented. To invoke other Smart Contracts methods participants will need to publish specially structured information to the smart space, which should contain:

1) the identifier of the Smart Contract whose method to be invoked;
2) the identifier of the method to be invoked;
3) the list of parameters;
4) other service parameters, specific to the certain implementation of the blockchain (for example, the amount of currency to be transferred in Ethereum).

The Smart Contract of the smart space will trace that information and carry out invocations of appropriate Smart Contracts methods. If the invocation fails the transaction of the blockchain (and all information published in it) will be rolled back. If the invocation succeeds the result will be added to the information about the invocation, and it will be published in the smart space (also, some information can be published by the invoked method). The invocation of Smart Contracts methods by publishing information in the smart space in a common format (for example, RDF) allows its participants to use only standard libraries and interfaces for working with smart space. The described scheme is shown in Fig. 3.


```
info = {
_:x rdf:type SCCall .
_:x :sc_name AssetsOwnership .
_:x :sc_method setOwner .
_:x :parameters [assetURI, newOwnerURI] .
}
```

```
info = {
assetURI :hasOwner prevOwnerURI delete .
assetURI :hasOwner newOwnerURI .
}
```

Fig. 3. A scheme of a third-party Smart Contract method invocation through the Smart Contract of the smart space

In addition to support of protocols over smart space, Smart Contracts can be used for other purposes. For example, act as a representative of a certain participant. The participant can create a Smart Contract that will publish information in the smart space on his behalf. An example may be an enterprise that distributes certain physical resources that can be purchased and used in accordance with the terms provided. For example, an energy company can sell energy through a blockchain [8]. Purchased resources can be represented by virtual tokens, through which they can be controlled (in the simplest case – can be just used). For the distribution of tokens, a Smart Contract can be created, which on behalf of the enterprise can automatically create and transfer tokens (by publishing certain information to the smart space) at the direct request of the buyer. They can buy tokens for the currency build into the blockchain or for other resources (for example, bank bonds), which can also be objects of the smart space. The exchange of resources must take place atomically, so it is necessary that a Smart Contract can publish information also on behalf of the participant who invoked it.

So, to support the described use cases, the Smart Contract of the smart space should support:

1) The invocation of other Smart Contracts of the blockchain through the publication of the specially structured information.
2) The publication of information by other Smart Contract on behalf of itself, his creator and the transaction initiator.

In addition, to solve the stated task of developing a platform for the IIoT, another interesting application of the Smart Contract in the smart space should be noted. It can be used to combine heterogeneous states of several independent Smart Contracts in one smart space. That allows, for example, to perform a joint search and conditions checks on it. This application can be discussed.

B. Architecture

One of the functional requirements to the smart space platform is the possibility of a complex search for the stored information, but the implementation of the blockchain technology limits us in this. The blockchain can be viewed as an unstructured transaction journal with the ability to store the state, providing for this a limited set of data structures types. This set limits the search capabilities, for example, in the default build of Hyperledger [22], the key-value database is used to store the state. This problem can be solved by creating an additional layer on top of the blockchain (Fig. 4). This layer will read the transactions journal and create its own state, while any structures that support the necessary search capabilities can be used to store the data (for example, the SPARQL query language for data in the triplet format). The blockchain will be used as a means of information distribution, consensus building, users managing, conducting basic checks of syntax and semantics as well as representing certain capabilities (for example, currency management in Ethereum). In this layer additional checks of consistency can be made (in accordance with the accepted semantics) to produce which in Smart Contracts of the blockchain would be inefficient.

The layer can store the local information of the participant, which will not be published in the blockchain, but will complement public information. For example, in a smart factory a single smart space can be used for communication of local machines among themselves and with remote services, machines of other factories and so on. However, information, necessary for local communication is not required to be published in the blockchain.

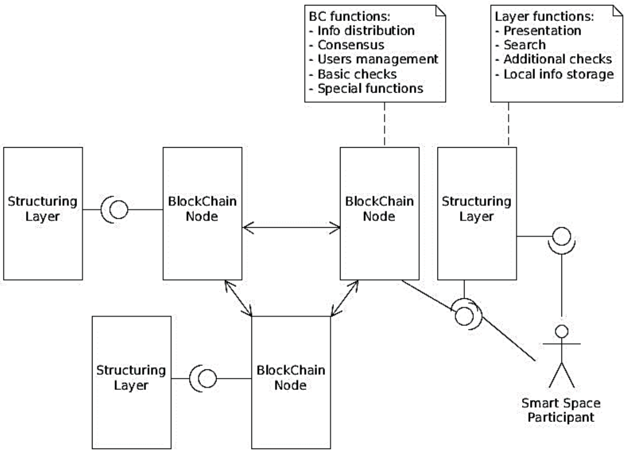


Fig. 4. System architecture of the blockchain-based platform for IIoT

For the organization of smart space, a promising platform is Smart-M3. Its usage on the IoT implementation is described in many papers, for example [23],[3] including the IIoT [12]. The system architecture of the solution is shown in Fig. 5.

To integrate Smart-M3 platform with the blockchain a component can be developed that will read the transaction journal and produce the state changes in the Smart-M3-based smart space using the standard Smart-M3 interface. To search for and subscribe to information in the smart space, local KPs can send requests directly to the Smart-M3, but the requests for update must be addressed to the integration component that will redirect them to the blockchain. The response to the KP will be sent and the change in the state of the Smart-M3-based smart space will occur only when changes are accepted in the blockchain (after the emergence of the next transactions block, which contains those changes). It should be noted that the request for update can be rejected if the changes violate the consistency rules of the blockchain smart space (also if the request initiated an invocation of third-party Smart Contract method that resulted in an error), in this case KP-initiator will receive an error response.

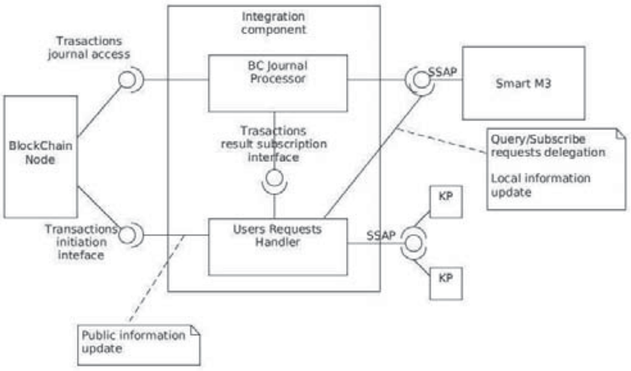

Integration component
Trasactions journal access
BC Journal Processor
SSAP
Smart M3
BlockChain Node
Trasactions result subscription interface
Query/Subscribe requests delegation
Local information update
KP
Transactions initiation inteface
Users Requests Handler
SSAP
KP
Public information update

Fig. 5. A system architecture of integration of the blockchain with the Smart-M3 platform

The requests to write local information that do not need to be published in the blockchain can be sent directly to the Smart-M3 while KPs will decide by themselves which information should be published in the blockchain (through the integration component). The possible alternative can be to send all requests for update to the central integration component in which filters will be set to separate information that must be published in the blockchain from local information. Local information will be written into the Smart-M3-based smart space immediately, the public information will be sent to the blockchain first. An advantage of this solution is in the centralized management of information visibility.

The presence of an integration component may not be noticeable for KPs. For this, the component must implement the same protocol as Smart-M3. It can accept all requests (queries and updates) from KP, while KP will consider that they are communicating with Smart-M3. This solution will make it possible to use existing client libraries to work with the smart space of Smart-M3, for example described in the paper [23].

However, the use of blockchain has certain specificities that must be considered when developing a KP:

1) Changes in the smart space will be made only after the emergence of the new transactions block containing those changes, which can occur with a long delay, which depends on the implementation of the blockchain and requests waiting to be processed. For example, in Ethereum a new block is formed approximately every 17 seconds, while there is a restriction on the number of transactions it can contain. In this regard Hyperledger is better – a new block is formed almost immediately after the request is processed (the strategy can be configured).
2) A request for an update may fail if the changes violate smart space consistency rules.
3) In some implementations of the blockchain, the state can be rolled back if alternative chain of blocks appears which should be accepted in accordance with the consensus algorithm. Such situation should occur rarely, but it can. Possible solutions:
    - The development of KPs with possibility of a smart space state rollback, which will be perceived as a standard situation.
    - The processing of such situation as an emergency, carrying out rigid restorative measures to continue work. For example, the production of the current goods in the factory can be stopped and returned to the beginning of the process. This solution can be acceptable in some cases if the rollback of the blockchain state happens rare. To reduce the likelihood of such situation, before the processing of the next transactions block the integration component can wait an additional timeout.
    - Use of the blockchain implementation in which the state cannot be rolled back.

Another feature of the integration component can be the filtering of information from the blockchain, which should be written into the Smart-M3-based smart space, for example, according to its author. One blockchain network can be used to conduct many different business processes, but a specific participant may only need information of participants of the same business processes.

VI. Discussion

Several advantages and disadvantages can be distinguished in the proposed architecture. The advantages are related to the use of blockchain jointly with the IoT platform that makes it possible to smooth out the shortcomings of each of them. Thus, the use of blockchain allows to provide mechanisms for ensuring the trust, durability of storage and non-repudiation from information, as well as a consensus mechanism, the implementation of which by the means of the used IoT platform would require considerable effort.

In turn, the use of IoT in jointly with the blockchain allows to present information transferred through blockchain using ontologies. At the same time, there is a slight duplication of information, but it allows to provide search, including semantic search, in blocks of blockchain transactions.

The mechanism of smart contracts used in the architecture allows to determine the conditions and actions of any complexity, which in turn makes it possible to set conditions and reactions to conditions quite complex. Here is also could be some problems related to the complexity of contract. The computational complexity of the contract can be so great that it will never be fulfilled. However, this problem can be solved due to the specific of the blockchain technology. When checking a block, static code analysis can be performed and potentially dangerous code will be isolated and rejected.

The immutability of the blocks and the storage of the transactions log is not only an advantage, but also a disadvantage. It is related to the fact that the constantly growing chain of blocks requires a significant amount of memory, which is poorly consistent with the simple devices that works in the IoT. However, it is assumed that this problem can be solved by using ontologies processed on a more powerful device, whereby weak devices can delegate some of their information and functions for working with blockchain to other devices in the smart space.

The other possible disadvantage is related to the input of information into the smart space that is carried out with a delay because of the need to form and agree a block of transactions before entering information. This problem can be solved by selecting and configuring the environment in such a way that it takes as little time as possible to form new blocks.

Due to the specific of the blocks creation in the blockchain, an alternative branch may arise, which should be adopted in accordance with the consensus algorithm. To solve this problem, special procedures are developed in the blockchain platforms, which allow to resolve such discrepancies while preserving the original chain of blocks.

In the future work, it is planned to develop an ontology for the description of a smart contract that will provide interoperability for the interaction between several smart spaces. It is planned to find solutions to the problems presented above and to supplement them with the proposed architecture. In addition, it is planned to implement a research prototype to test the speed and correctness of the proposed architecture.

VII. Conclusion

The current level of information technology development allows to assert that the production process has entered a new stage — Industry 4.0. There is a gradual automation of production, the introduction of additive printing technologies, the development of artificial intelligence, IoT, giving production robots greater freedom of action depending on the current situation. Altogether, it allows to provide high customization of the final product without the need for a strong manufacture reorganization.

To achieve this goal, the interaction between the components of production is carried out based on the concept of Internet of Things. Analysis of existing IoT platforms has shown that they have several shortcomings, among which there are lack of mechanisms for establishing authorship, durability and unchangeability of information, control over the exchange of resources in production, and an integrated mechanism for reaching consensus among participants.

To solve the above problems an architecture is proposed that combine IoT and blockchain technology. Such combination made it possible to use the mechanisms implemented in the blockchain to solve the problems identified in the platforms for IoT. To control resources in the production process, it is suggested to use a smart contract in the work that determines the conditions for the transfer of the resources, and possible operations with resources in the smart space.

The analysis of the proposed solution has made it possible to determine the several of its advantages, among which one can single out the search for information in the detachment, the mechanisms for achieving consensus and the unchangeability of information available through the IoT platform. Among the shortcomings can be identified a potentially large size of the chain of blocks and a reduction in the speed of information entry into the smart space due to the calculation of the transaction block.

The future work will be aimed at elimination of the listed shortcomings and practical implementation of the proposed architecture.

ACKNOWLEDGMENT

The research is funded by the Russian Science Foundation (Project № 17-71-10223).

REFERENCES

[1] K. Schwab, *The fourth industrial revolution*. World Economic Forum, 2016.

[2] H. Lasi, P. Fettke, H. G. Kemper, T. Feld, and M. Hoffmann, “Industry 4.0,” *Bus. Inf. Syst. Eng.*, vol. 6, no. 4, pp. 239–242, 2014.

[3] A. Smirnov, A. Kashevnik, A. Ponomarev, N. Shilov, M. Shchekotov, and N. Teslya, “Smart space-based intelligent mobile tourist guide: Service-based implementation”, *Conference of Open Innovation Association, FRUCT*, 2014, pp. 126–134.

[4] A. Smirnov, A. Kashevnik, N. Shilov, S. Balandin, I. Oliver, and S. Boldyrev, “On-the-fly ontology matching in smart spaces: A multi-model approach,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6294, pp. 72–83, 2010.

[5] O. Goldreich, *Foundations of cryptography I: Basic Tools*, Cambridge: Cambridge University Press, 2001.

[6] A. Smirnov, A. Kashevnik, N. Shilov, and N. Teslya, “Context-Aware Access Control Model for Privacy Support in Mobile-Based Assisted Living”, *J. Intell. Syst.*, vol. 24, no. 3, pp. 333–342, 2015.

[7] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” Www.Bitcoin.Org, p. 9, 2008.

[8] J. J. Sikorski, J. Haughton, M. Kraft, P. Street, and P. F. Drive, “Blockchain technology in the chemical industry: machine-to-machine electricity market,” *Appl. Energy*, vol. 195, no. 178, pp. 234–246, Jun. 2016.

[9] S. A. Abeyratne and R. P. Monfared, “Blockchain ready manufacturing supply chain using distributed ledger,” *Int. J. Res. Eng. Technol.*, vol. 5, no. 9, pp. 1–10, Sep. 2016.

[10] S. El Kadiri, B. Grabot, K.-D. Thoben, K. Hribernik, C. Emmanouilidis, G. von Cieminski, and D. Kiritsis, “Current trends on ICT technologies for enterprise information systems,” *Comput. Ind.*, vol. 79, pp. 14–33, Jun. 2016.

[11] X. Jia, R. A. Fathy, Z. Huang, S. Luo, J. Gong, and J. Peng, *Framework of blockchain of things as decentralized service platform*, 2017.

[12] A. Kashevnik, N.Teslya, E. Yablochnikov, V. Arckhipov and K. Kipriyanov, Hybrid automated line workstations interaction scenario for optical devices assembly 2016 18th Conference of Open Innovations Association and Seminar on Information Security and Protection of Information Technology (FRUCT-ISPIT), St. Petersburg, 2016, pp. 92-99.

[13] J. Honkola, H. Laine, R. Brown, and O. Tyrkkö, “Smart-M3 information sharing platform,” in *Proceedings - IEEE Symposium on Computers and Communications*, 2010, pp. 1041–1046.

[14] W. Xie, Y. Shi, G. Xu, and Y. Mao, “Smart Platform - a software infrastructure for Smart Space (SISS),” in *Proceedings. Fourth IEEE International Conference on Multimodal Interfaces*, pp. 429–434.

[15] J. Aubert, C. Feltus, A. Kostakis, and D. Khadraoui, “Smart-X: an Adaptive Multi-Agent Platform for Smart-Topics,” *Procedia Comput. Sci.*, vol. 109, pp. 943–948, Jan. 2017. *1*

[16] T. Zhu, S. Dhelim, Z. Zhou, S. Yang, and H. Ning, “An architecture for aggregating information from distributed data nodes for industrial internet of things,” *Comput. Electr. Eng.*, vol. 58, pp. 337–349, Feb. 2017.

[17] S. Boldyrev, I. Oliver, and J. Honkola, “A mechanism for managing and distributing information and queries in a smart space environment,” *1st Int. Work. Manag. Data with Mob. Devices (MDMD 2009)(Milan, Italy)*, pp. 1–10, 2009.

[18] D. Drescher, *Blockchain basics: a non-technical introduction in 25 steps*. 2017.

[19] M. B. Hoy, “An Introduction to the Blockchain and Its Implications for Libraries and Medicine,” *Med. Ref. Serv. Q.*, vol. 36, no. 3, pp. 273–279, Jul. 2017.

[20] N. Szabo, “Smart Contracts: Building Blocks for Digital Markets Copyright,” alamut.com, 1996. Web: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html. [Accessed: 16-Sep-2017].

[21] Ethereum Web: https://www.ethereum.org/ [Accessed: 16-Sep-2017]

[22] Hyperledger Fabric Web: https://hyperledger-fabric.readthedocs.io/ [Accessed: 16-Sep-2017]

[23] D. G. Korzun, S. I. Balandin, V. Luukkala, P. Liuha, and A. V. Gurtov, “Overview of Smart-M3 principles for application development,” *Proc. Int. Conf. Artif. Intell. Intell. Syst.*, 2011.